

[illegible]LN  
LN  
LNLN  
LN  
LN

LN  
LN  
LN  
LN  
LN  
LN

LN

LN

LN  
LNLN  
LNLN  
LNLN  
LNLN  
LNLN  
LN

LN

LN

LN

LN  
LNLN  
LN

LN

100

100

```
LL      NN      NN      KK      KK      VV      VV      MM      MM      CCCCCCCC  TTTTTTTTTT  RRRRRRRR  LL
LL      NN      NN      KK      KK      VV      VV      MM      MM      CCCCCCCC  TTTTTTTTTT  RRRRRRRR  LL
LL      NN      NN      KK      KK      VV      VV      MM      MM      CC          TT          RR          LL
LL      NN      NN      KK      KK      VV      VV      MM      MM      CC          TT          RR          LL
LL      NN      NN      KK      KK      VV      VV      MM      MM      CC          TT          RR          LL
LL      NN      NN      KK      KK      VV      VV      MM      MM      CC          TT          RR          LL
LL      NN      NN      KK      KK      VV      VV      MM      MM      CC          TT          RR          LL
LL      NN      NN      KK      KK      VV      VV      MM      MM      CC          TT          RR          LL
LL      NN      NN      KK      KK      VV      VV      MM      MM      CC          TT          RR          LL
LL      NN      NN      KK      KK      VV      VV      MM      MM      CC          TT          RR          LL
LL      NN      NN      KK      KK      VV      VV      MM      MM      CC          TT          RR          LL
LL      NN      NN      KK      KK      VV      VV      MM      MM      CC          TT          RR          LL
LLLLLLLLLLLL  NN      NN      KK      KK      VV      VV      MM      MM      CCCCCCCC  TTTTTTTTTT  RRRRRRRR  LLLLLLLLLL
LLLLLLLLLLLL  NN      NN      KK      KK      VV      VV      MM      MM      CCCCCCCC  TTTTTTTTTT  RRRRRRRR  LLLLLLLLLL
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLLLL  IIIIII  SSSSSSSS
```

TR  
SE  
IM



```
1 0001 0 MODULE LNK_VMCTRL ( ! CONTROL FOR ALLOCATION OF VIRTUAL MEMORY
2 0002 0 IDENT = 'V04-000',
3 0003 0 ADDRESSING_MODE(EXTERNAL=GENERAL,
4 0004 0 NONEXTERNAL=LONG_RELATIVE)
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY: LINKER
33 0033 1
34 0034 1 ABSTRACT: THIS MODULE CONTAINS THE ROUTINES TO ALLOCATE VIRTUAL MEMORY
35 0035 1 AT END OF PASS 1.
36 0036 1
37 0037 1
38 0038 1 ENVIRONMENT: VMS NATIVE MODE
39 0039 1
40 0040 1 AUTHOR: T.J. PORTER, CREATION DATE: 15-JUN-78
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 V03-001 BLS0007 Benn Schreiber, 3-Jun-1980
45 0045 1 Convert to MDL data structures.
46 0046 1 --
```

```

48      0047 1 |
49      0048 1 |++
50      0049 1 |      FUNCTIONAL DESCRIPTION
51      0050 1 |
52      0051 1 | LIBRARY
53      0052 1 | 'STARLETL32';
54      0053 1 | REQUIRE
55      0054 1 | 'PREFIX';
56      0169 1 | LIBRARY
57      0170 1 | 'DATBAS';
58      0171 1 |
59      0172 1 | EXTERNAL ROUTINE
60      0173 1 |          LNK$ALLOBLK : NOVALUE,          ! DYNAMIC MEMORY ALLOCATOR
61      0174 1 |          LNK$DEALBLK : NOVALUE;          ! AND DEALLOCATOR
62      0175 1 |
63      0176 1 | EXTERNAL
64      0177 1 |          LNK$GL_FVMLST;                  ! LISTHEAD OF FREE VIR MEM DESCRIPTORS
65      0178 1 |
66      0179 1 | GLOBAL
67      0180 1 |          LNK$GL_MINVA : INITIAL(MAX_ADDRESS), ! LOWEST ADDRESS ALLOCATED
68      0181 1 |          LNK$GL_MAXVA : INITIAL(0);        ! HIGHEST ADDRESS ALLOCATED

```



```
0182 1 GLOBAL ROUTINE LNK$ALLOVIRMEM(ADDRESS,PAGES) =
0183 2 BEGIN
0184 2
0185 2 ++
0186 2 THIS ROUTINE IS CALLED TO ATTEMPT ALLOCATION OF A SPECIFIC
0187 2 PIECE OF VIRTUAL MEMORY. THE ADDRESS AND PAGE COUNT REQUIRED
0188 2 ARE THE INPUT ARGUMENTS. THE ROUTINE RETURNS TRUE OR FALSE
0189 2 DEPENDING ON WHETHER THE SPECIFIED MEMORY IS AVAILABLE OR
0190 2 ALREADY ALLOCATED.
0191 2
0192 2 THE AVAILABLE VIRTUAL MEMORY IS CONTROLLED BY A SINGLY LINKED
0193 2 LIST OF BLOCKS WITH FORMAT:
0194 2 -----
0195 2 ! NEXT DESCRIPTOR !
0196 2 -----
0197 2 ! FREE ADDRESS !
0198 2 -----
0199 2 ! # FREE BYTES !
0200 2 -----
0201 2 --
0202 2
0203 2 LOCAL
0204 2 BYTES,                                ! BYTE COUNT REQUESTED
0205 2 FREBLK : REF BLOCK[.BYTE],           ! FREE BLOCK POINTER
0206 2 PREVBLK : REF BLOCK[.BYTE],         ! PREVIOUS FREE BLOCK IN LIST
0207 2
0208 2 IF (BYTES = .PAGES * 512) EQL 0      ! IF ZERO SIZE
0209 2 THEN RETURN TRUE;                  ! RETURN SUCCESS
0210 2
0211 2 PREVBLK = LNK$GL_FVMLST;             ! SET AT LISTHEAD
0212 2 WHILE (FREBLK = .PREVBLK[FVMSL_NXTFVM]) NEQ 0 ! MOVE TO NEXT BLOCK DESCRIPTOR IN LIST
0213 2 DO IF .ADDRESS LSSU .FREBLK[FVMSL_ADDRESS] ! LOOP TILL WE GET TO ONE FOR REQUESTED MEMORY
0214 2 OR .ADDRESS GEQU (.FREBLK[FVMSL_ADDRESS] + .FREBLK[FVMSL_BYTES])
0215 2 THEN PREVBLK = .FREBLK             ! ...
0216 2 ELSE BEGIN
0217 2 IF (.ADDRESS + .BYTES) GTRU (.FREBLK[FVMSL_ADDRESS] + ! DOES DESIRED BLOCK FIT
0218 2 .FREBLK[FVMSL_BYTES])              ! IN THE HOLE
0219 2 THEN EXITLOOP;                     ! QUIT IF NOT
0220 2 IF .ADDRESS EQL .FREBLK[FVMSL_ADDRESS] ! IF AT FRONT OF HOLE
0221 2 THEN IF (FREBLK[FVMSL_BYTES] = .FREBLK[FVMSL_BYTES] - ! REDUCE THE SIZE OF REMAINDER
0222 2 .BYTES) EQL 0                      ! AND IF GOING TO ZERO
0223 2 THEN BEGIN
0224 2 PREVBLK[FVMSL_NXTFVM] = .FREBLK[FVMSL_NXTFVM]; ! TAKE DESCRIPTOR OFF LIST
0225 2 LNK$DEALBLK(FVMSL_SIZE,.FREBLK); ! AND DEALLOCATE IT
0226 2 END
0227 2 ELSE FREBLK[FVMSL_ADDRESS] = .FREBLK[FVMSL_ADDRESS] + .BYTES ! SET ADDRESS OF REMAINDER
0228 2 ELSE BEGIN                          ! NOT AT THE FRONT
0229 2 IF (.FREBLK[FVMSL_ADDRESS] + .FREBLK[FVMSL_BYTES]) EQL ! CHECK IF IT IS ON END
0230 2 (.ADDRESS + .BYTES)                ! OF THE BLOCK
0231 2 THEN FREBLK[FVMSL_BYTES] = .FREBLK[FVMSL_BYTES] - .BYTES ! IF SO JUST REDUCE SIZE
0232 2 ELSE BEGIN
0233 2 LOCAL NXTBLK : REF BLOCK[.BYTE]; ! OTHERWISE NEED TO SPLIT THE BLOCK
0234 2 LNK$ALLOBLK(FVMSL_SIZE,NXTBLK); ! ALLOCATE ANOTHER DESCRIPTOR
0235 2 NXTBLK[FVMSL_NXTFVM] = .FREBLK[FVMSL_NXTFVM]; ! WHICH POINTS ONWARD
0236 2 FREBLK[FVMSL_NXTFVM] = .NXTBLK; ! AND POINTED TO BY CURRENT
0237 2 NXTBLK[FVMSL_ADDRESS] = .ADDRESS + .BYTES; ! DESCRIBES THE REMNANT AT END
0238 2 NXTBLK[FVMSL_BYTES] = .FREBLK[FVMSL_BYTES] -
```

LNK\_VMCTRL  
V04=000

G 4  
16-Sep-1984 00:39:52 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:40:38 [LINKER.SRC]LNKVMCTRL.B32;1

Page 4  
(3)

```

: 127      0239 6
: 128      0240 5
: 129      0241 5
: 130      0242 4
: 131      0243 3
: 132      0244 3
: 133      0245 3
: 134      0246 3
: 135      0247 3
: 136      0248 3
: 137      0249 2
: 138      0250 2
: 139      0251 1

      (.NXTBLK[FVMSL_ADDRESS] -
      .FREBLK[FVMSL_ADDRESS]);
      FREBLK[FVMSL_BYTES] = .ADDRESS - .FREBLK[FVMSL_ADDRESS];      ! AND SET CURRENT FR
      END;
      END;
      IF (.ADDRESS + .BYTES - 1) GTRU .LNK$GL_MAXVA      ! MAXIMIZE THE ADDRESS
      THEN LNK$GL_MAXVA = .ADDRESS + .BYTES = 1;      ! SPACE ALLOCATED
      IF .ADDRESS - LSSU .LNK$GL_MINVA      ! AND RESET MINIMUM IF NEXCESSARY
      THEN LNK$GL_MINVA = .ADDRESS;
      RETURN TRUE;
      END;
      RETURN FALSE;
      END;
      ! OF LNK$ALLOVIRMEM
```

```

      .TITLE LNK_VMCTRL
      .IDENT \V04-000\
      .PSECT $GLOBAL$,NOEXE,2
      C0000000 00000 LNK$GL_MINVA::
      .LONG -1073741824
      00000000 00004 LNK$GL_MAXVA::
      .LONG 0
      .EXTRN LNK$ALLOBLK, LNK$DEALBLK
      .EXTRN LNK$GL_FVMLST
      .PSECT $CODE$,NOWRT,2
      .ENTRY LNK$ALLOVIRMEM, Save R2,R3,R4,R5,R6,R7,R8 : 0182
      MOVAB LNK$GL_MAXVA, R8
      SUBL2 #4, SP
      ASHL #9, PAGES, BYTES : 0208
      BNEQ 1$
      BRW 11$
      MOVAB LNK$GL_FVMLST, PREVBLK : 0211
      MOVL ADDRESS, R4 : 0213
      MOVL (PREVBLK), FREBLK : 0212
      BNEQ 3$
      BRW 12$
      CMPL R4, 4(FREBLK) : 0213
      BLSSU 4$
      ADDL3 8(FREBLK), 4(FREBLK), R1 : 0214
      CMPL R4, R1
      BLSSU 5$
      MOVL FREBLK, PREVBLK : 0215
      BRB 2$
      ADDL3 BYTES, R4, R5 : 0217
      MOVAB 8(FREBLK), R3 : 0218
      ADDL3 (R3), 4(FREBLK), R1
      CMPL R5, R1 : 0217
      BGTRU 12$
      CMPL R4, 4(FREBLK) : 0220
      BNEQ 7$
      SUBL2 BYTES, (R3) : 0222
      BNEQ 6$
```



LNK\_VMCTRL  
V04=000

H 4  
16-Sep-1984 00:39:52  
14-Sep-1984 12:40:38

VAX-11 Bliss-32 V4.0-742  
[LINKER.SRC]LNKVMCTRL.B32;1

Page 5  
(3)

		67		62	D0 0005C	MOVL	(FREBLK), (PREVBLK)	:	0224
				52	DD 0005F	PUSHL	FREBLK	:	0225
				0C	DD 00061	PUSHL	#12	:	
		00000000G	00	02	FB 00063	CALLS	#2, LNK\$DEALBLK	:	
				38	11 0006A	BRB	9\$	:	0221
		04	A2	56	C0 0006C 6\$:	ADDL2	BYTES, 4(FREBLK)	:	0227
				32	11 00070	BRB	9\$	:	0221
			55	51	D1 00072 7\$:	CMPL	R1, R5	:	0230
				05	12 00075	BNEQ	8\$	:	
			63	56	C2 00077	SUBL2	BYTES, (R3)	:	0231
				28	11 0007A	BRB	9\$	:	
				5E	DD 0007C 8\$:	PUSHL	SP	:	0234
				0C	DD 0007E	PUSHL	#12	:	
		00000000G	00	02	FB 00080	CALLS	#2, LNK\$ALLOBLK	:	
			50	6E	D0 00087	MOVL	NXTBLK, R0	:	0235
			60	62	D0 0008A	MOVL	(FREBLK), (R0)	:	
			62	50	D0 0008D	MOVL	R0, (FREBLK)	:	0236
		04	A0	55	D0 00090	MOVL	R5, 4(R0)	:	0237
		04	A2	04	A0 C3 00094	SUBL3	4(R0), 4(FREBLK), R1	:	0240
			51	63	C1 0009A	ADDL3	(R3), R1, 8(R0)	:	0239
			54	A2	C3 0009F	SUBL3	4(FREBLK), R4, (R3)	:	0241
08	51		50	A5	9E 000A4 9\$:	MOVAB	-1(R5), R0	:	0244
	A0		68	50	D1 000A8	CMPL	R0, LNK\$GL_MAXVA	:	
	63			03	1B 000AB	BLEQU	10\$	:	
			68	50	D0 000AD	MOVL	R0, LNK\$GL_MAXVA	:	0245
		FC	A8	54	D1 000B0 10\$:	CMPL	R4, LNK\$GL_MINVA	:	0246
				04	1E 000B4	BGEQU	11\$	:	
		FC	A8	54	D0 000B6	MOVL	R4, LNK\$GL_MINVA	:	0247
			50	01	D0 000BA 11\$:	MOVL	#1, R0	:	0248
					04 000BD	RET		:	
				50	D4 000BE 12\$:	CLRL	R0	:	0250
				04	000C0	RET		:	0251

; Routine Size: 193 bytes, Routine Base: \$CODE\$ + 0000

```
141 0252 1 GLOBAL ROUTINE LNK$FINDVIRMEM(RETADR,PAGES,LOWESTVA) =
142 0253 2 BEGIN
143 0254 2
144 0255 2 ++
145 0256 2 THIS ROUTINE IS SIMILAR TO LNK$ALLOVIRMEM EXCEPT THAT ANY FREE
146 0257 2 VIRTUAL MEMORY LARGE ENOUGH IS ALLOCATED, STARTING AT LOWEST
147 0258 2 ADDRESS END.
148 0259 2
149 0260 2 IF LOWESTVA IS SUPPLIED IT SPECIFIES THAT THE VIRTUAL MEMORY
150 0261 2 ALLOCATED MUST BE AT A HIGHER ADDRESS THAN LOWESTVA
151 0262 2
152 0263 2 RETADR = ADDRESS OF CELL TO RECEIVE THE ADDRESS OF
153 0264 2 THE FIRST BYTE ALLOCATED.
154 0265 2
155 0266 2 --
156 0267 2 MAP
157 0268 2 RETADR : REF VECTOR[,LONG];
158 0269 2
159 0270 2 BUILTIN
160 0271 2 NULLPARAMETER;
161 0272 2
162 0273 2 LOCAL
163 0274 2 BYTES,
164 0275 2 FREBLK : REF BLOCK[,BYTE],
165 0276 2 PREVBLK : REF BLOCK[,BYTE],
166 0277 2 SPECIALADR;
167 0278 2
168 0279 2 IF (BYTES = .PAGES * 512) EQL 0
169 0280 2 THEN BEGIN
170 0281 3 RETADR[0] = 0; ! IF REQUESTED SIZE IS ZERO
171 0282 3 RETURN TRUE ! SUCCESS
172 0283 2 END;
173 0284 2
174 0285 2 SPECIALADR = NOT NULLPARAMETER(3) ! SET FLAG IF TO ALLOCATE ABOVE SPECIFIC ADDRESS
175 0286 2 AND (.LOWESTVA NEQ 0);
176 0287 2
177 0288 2 PREVBLK = LNK$GL_FVMLST; ! SET TO START OF LIST
178 0289 2 WHILE (FREBLK = .PREVBLK[FVMSL_NXTFVM]) NEQ 0 ! GO DOWN LIST LOOKING FOR LARGE ENOUGH PIECE OF FRE
179 0290 3 DO IF (IF .SPECIALADR
180 0291 5 THEN ((.FREBLK[FVMSL_ADDRESS] LSSU .LOWESTVA)
181 0292 6 OR ((.FREBLK[FVMSL_ADDRESS]+.FREBLK[FVMSL_BYTES]
182 0293 6 LSSU .LOWESTVA)
183 0294 6 OR (.FREBLK[FVMSL_ADDRESS]+.FREBLK[FVMSL_BYTES]
184 0295 4 LSSU .LOWESTVA+.BYTES)))
185 0296 3 ELSE (.BYTES GTRU .FREBLK[FVMSL_BYTES]))
186 0297 2 THEN PREVBLK = .FREBLK
187 0298 2 ELSE BEGIN
188 0299 3 RETADR[0] = .FREBLK[FVMSL_ADDRESS]; ! RETURN ITS VIRTUAL ADDRESS
189 0300 4 IF (FREBLK[FVMSL_BYTES] = .FREBLK[FVMSL_BYTES] -- ! REDUCE ITS SIZE
190 0301 3 .BYTES) EQL 0 ! AND IF COMPLETELY CONSUMED
191 0302 4 THEN BEGIN
192 0303 4 PREVBLK[FVMSL_NXTFVM] = .FREBLK[FVMSL_NXTFVM]; ! REMOVE DESCRIPTOR FROM LIST
193 0304 4 LNK$DEALBLK(FVMSL_SIZE,.FREBLK); ! AND DEALLOCATE THE DESCRIPTOR
194 0305 4 END
195 0306 3 ELSE FREBLK[FVMSL_ADDRESS] = .FREBLK[FVMSL_ADDRESS] + ! OTHERWISE JUST ADJUST THE
196 0307 3 .BYTES; ! VIRTUAL ADDRESS REMAINING
197 0308 3 IF (.RETADR[0] + .BYTES - 1) GTRU .LNK$GL_MAXVA ! MAXIMIZE THE ADDRESS
```



```

: 198      0309 3      THEN LNK$GL_MAXVA = ..RETADR + .BYTES - 1;      ! SPACE ALLOCATED
: 199      0310      IF .RETADR[0] LSSU .LNK$GL_MINVA      ! AND RESET MINIMUM IF NEXCESSARY
: 200      0311      THEN LNK$GL_MINVA = .RETADR[0];
: 201      0312      RETURN TRUE      ! RETURN SUCCESS
: 202      0313      END;
: 203      0314 2 RETURN FALSE      ! NO MEMORY FOUND
: 204      0315 1 END;      ! END OF LNK$FINDVIRMEM
```

53	08	56	00000000'	007C 00000	.ENTRY LNK\$FINDVIRMEM, Save R2,R3,R4,R5,R6	0252
		AC		EF 9E 00002	MOVAB LNK\$GL_MAXVA, R6	
				09 78 00009	ASHL #9, PAGES, BYTES	0279
			04	06 12 0000E	BNEQ 1\$	
				BC D4 00010	CLRL @RETADR	0281
			0093	31 00013	BRW 12\$	0282
		03		6C 91 00016 1\$:	CMPB (AP), #3	0285
		50		05 1E 00019	BGEQU 2\$	
				01 D0 0001B	MOVL #1, R0	
				09 11 0001E	BRB 3\$	
				50 D4 00020 2\$:	CLRL R0	
			0C	AC D5 00022	TSTL 12(AP)	
				02 12 00025	BNEQ 3\$	
				50 D6 00027	INCL R0	
				51 D4 00029 3\$:	CLRL R1	0286
			0C	AC D5 0002B	TSTL LOWESTVA	
				02 13 0002E	BEQL 4\$	
				51 D6 00030	INCL R1	
55		51		50 CB 00032 4\$:	BICL3 R0, R1, SPECIALADR	
		54	00000000G	00 9E 00036	MOVAB LNK\$GL_FVMLST, PREVBLK	0288
		52		64 D0 0003D 5\$:	MOVL (PREVBLK), FREBLK	0289
				6B 13 00040	BEQL 13\$	
		1F		55 E9 00042	BLBC SPECIALADR, 6\$	0291
	0C	AC	04	A2 D1 00045	CMPL 4(FREBLK), LOWESTVA	
				1E 1F 0004A	BLSSU 7\$	
51	04	A2	08	A2 C1 0004C	ADDL3 8(FREBLK), 4(FREBLK), R1	0292
	0C	AC		51 D1 00052	CMPL R1, LOWESTVA	0293
				12 1F 00056	BLSSU 7\$	
50		53	0C	AC C1 00058	ADDL3 LOWESTVA, BYTES, R0	0295
		50		51 D1 0005D	CMPL R1, R0	
				0D 1E 00060	BGEQU 8\$	
				06 11 00062	BRB 7\$	
	08	A2		53 D1 00064 6\$:	CMPL BYTES, 8(FREBLK)	0296
				05 1B 00068	BLEQU 8\$	
		54		52 D0 0006A 7\$:	MOVL FREBLK, PREVBLK	0297
				CE 11 0006D	BRB 5\$	
	04	BC	04	A2 D0 0006F 8\$:	MOVL 4(FREBLK), @RETADR	0299
	08	A2		53 C2 00074	SUBL2 BYTES, 8(FREBLK)	0301
				10 12 00078	BNEQ 9\$	
		64		62 D0 0007A	MOVL (FREBLK), (PREVBLK)	0303
				52 DD 0007D	PUSHL FREBLK	0304
				0C DD 0007F	PUSHL #12	
	00000000G	00		02 FB 00081	CALLS #2, LNK\$DEALBLK	
				04 11 00088	BRB 10\$	0300
	04	A2		53 C0 0008A 9\$:	ADDL2 BYTES, 4(FREBLK)	0307

LNK\_VMCTRL  
V04=000

K 4  
16-Sep-1984 00:39:52  
14-Sep-1984 12:40:38

VAX-11 Bliss-32 V4.0-742  
[LINKER.SRC]LNKVMCTRL.B32;1

Page 8  
(4)

	50	04	BC	D0	0008E	10\$:	MOVL	@RETADR, R0	:	0308
	51	FF	A340	9E	00092		MOVAB	-1(BYTES)[R0], R1	:	
	66		51	D1	00097		CMPL	R1, LNK\$GL_MAXVA	:	
			03	1B	0009A		BLEQU	11\$	:	
	66		51	D0	0009C		MOVL	R1, LNK\$GL_MAXVA	:	0309
FC	A6		50	D1	0009F	11\$:	CMPL	R0, LNK\$GL_MINVA	:	0310
			04	1E	000A3		BGEQU	12\$	:	
FC	A6		50	D0	000A5		MOVL	R0, LNK\$GL_MINVA	:	0311
	50		01	D0	000A9	12\$:	MOVL	#1, R0	:	0312
				04	000AC		RET		:	
			50	D4	000AD	13\$:	CLRL	R0	:	0314
				04	000AF		RET		:	0315

; Routine Size: 176 bytes, Routine Base: \$CODE\$ + 00C1

; 205 0316 0 END ELUDOM

#### PSECT SUMMARY

Name	Bytes	Attributes
\$GLOBALS	8	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	369	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

#### Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	6	0	581	00:01.0
\$255\$DUA28:[LINKER.OBJ]DATBAS.L32;1	538	4	0	28	00:00.5

#### COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:LNKVMCTRL/OBJ=OBJ\$:LNKVMCTRL MSRC\$:LNKVMCTRL/UPDATE=(ENH\$:LNKVMCTRL)

; Size: 369 code + 8 data bytes  
; Run Time: 00:09.7  
; Elapsed Time: 00:29.0  
; Lines/CPU Min: 1962  
; Lexemes/CPU-Min: 17180  
; Memory Used: 94 pages



LNK\_VMCTRL  
V04=000

L<sup>4</sup>  
16-Sep-1984 00:39:52

VAX-11 Bliss-32 V4.0-742

Page 9

; Compilation Complete

\_\$2

Vir  
Sta  
Ima  
Ima  
Ima  
Num  
Num  
Num  
Num  
Num  
Num  
Num  
Ima  
Map  
Est

Per  
---

Tot  
Usi  
Tot

Num

85

A t  
LIN



0220

AH-BT13A-SE  
 VAX/VMS V4.0

**DIGITAL EQUIPMENT CORPORATION**  
**CONFIDENTIAL AND PROPRIETARY**